REMARKS/ARGUMENTS

Claims 1-5, 14, and 21-22 are pending in the present application. Claims 6-13 and 15-20 are canceled; claims 1, 4, and 5 are amended; and claims 21-22 are added. Applicants are not conceding in this application that the claims as presented prior to this Amendment are not patentable over the art cited by the Examiner, as the present claim amendments are only for facilitating expeditious issuance of the application. Applicants respectfully reserve the right to pursue these and other claims in one or more continuations and/or divisional patent applications. Support for the amendments to the claims is located at least in the previous drafts of the claims and in the specification on page 7, line 9, through page 8, line 25; and in **Figure 3**. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 102, Anticipation

The Examiner has rejected claims 1, 2, 5, 6 and 14 under 35 U.S.C. § 102 as being anticipated by Fresko et al., U.S. Patent No. 5,966,702 (hereinafter "*Fresko*"). This rejection is respectfully traversed.

In rejecting the claims, the Examiner states:

> Regarding Claims 1, Fresko teaches: (Currently amended) A data processing method for creating an executable file by combining a plurality of run units, the method comprising the ~~steps of: reading a first run unit to be added to the executable file; locating a first data entity set to a first string value in the first run unit;~~ responsive to a determination that a first run unit to be added to the executable file comprises a first data entity set to a first value indicating that the first data entity is required to appear only once in the executable file (**Column 9, Lines 17-21, "In step 402, the pre-processor examines the constant pool tables of each class to determine the set of class file constants (such as strings and numerics, as well as others specific to the class file format) that can be shared between classes in "S."" i.e. determines which entities are "constants"**), determining whether the first data entity matches [[with]] a second data entity set to a second value and included in a second run unit, the second data entity being from a wherein the second run unit comprises a run unit that was previously added to the executable file (**Column 9, Lines 21 -23, "A shared constant pool table is created in step 403, with all duplicate constants determined from step 402."**); [[and]] responsive to a determination that the first data entity matches the second data entity, adding the first run unit to the executable file [[but]] without the first data entity (**Column 9, Lines 23-35, "In step 404, the pre-processor removes the duplicate, shared constants from the individual constant pool tables of each class."**). ; and responsive to a determination that the first data entity does not match the second data entity, adding the first run unit to the executable file with the first data entity (**Column 9, Lines 23-35, "In step 404, the pre-processor removes the duplicate, shared constants from the individual constant pool tables of each class." i.e. if they are NOT duplicates, they are NOT removed from the constant pool of the class when the class file is added**).
>
> Regarding Claims 2, Fresko teaches: (Currently amended) A method of claim 1 wherein ~~the step of matching matches~~ the first data entity [[with]] matches the second data entity if the first value and second value are identical. (**Column 9, Lines 21-23, "A**

shared constant pool table is created in step 403, with all duplicate constants determined from step 402.").

Regarding Claims 5, Fresko teaches:(Currently amended) A method of claim 1 wherein the determination that the first run unit to be added to the executable file comprises a first data entity set to a first value indicating that the first data entity is required to appear only once in the executable file **(Column 9, Lines 17-21, "In step 402, the pre-processor examines the constant pool tables of each class to determine the set of class file constants (such as strings and numerics, as well as others specific to the class file format) that can be shared between classes in "S."" i.e. determines which entities are "constants")**, the step of locating a first data entity comprises ~~the steps of~~: locating ~~two or more~~ a plurality of data entities in the first run unit **(Column 9, Lines 21-23, "A shared constant pool table is created in step 403, with all duplicate constants determined from step 402.")**; and creating the first data entity from the two ~~or more data~~ plurality of data entities **(Column 9, Lines 21-23, "A shared constant pool table is created in step 403, with all duplicate constants determined from step 402.")**.

Regarding Claim 6, Fresko teaches: (Currently amended) A method of claim 1 wherein the ~~step of locating a first data entity locates the first data entity using a key value by which the~~ first ~~data entity is marked~~ value is a key value **(Column 9, Lines 55-57, "In one embodiment of the invention, a new constant type is defined with a corresponding constant type tag. The new constant type provides as its info[] element an index into the shared constant table.")**.

Regarding Claim 14, Fresko teaches: (Currently amended) A method of claim 5 wherein the ~~step of~~ locating ~~two or more~~ a plurality of data entities comprises locating a plurality of data entities using a key value by which each of the ~~two or more~~ plurality of data entities is marked. **(Col. 9 Ln 55-57, "In one embodiment of the invention, a new constant type is defined with a corresponding constant type tag. The new constant type provides as its info[] element an index into the shared constant table.")**

Final Office Action dated August 22, 2008, pp. 2-4.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). In this case each and every feature of the presently claimed invention is not identically shown in *Fresko*, arranged as they are in the claims, and, accordingly, *Fresko* does not anticipate the claims. Specifically, *Fresko* does not teach or suggest "identifying a first data entity and a second data entity, wherein both the first data entity and the second data entity are identified using an Assembler instruction, and wherein the Assembler instruction identifies character strings which are required to appear only once in the executable file," as recited in amended independent claim 1.

Fresko is directed to a method and apparatus for pre-processing and packaging class files. Embodiments remove duplicate information elements from a set of class files to reduce the size of individual class files and to prevent redundant resolution of the information elements. Memory allocation requirements are determined in advance for the set of classes as a whole to reduce the complexity of memory allocation when the set of classes are loaded. The class files are stored in a single package for efficient storage, transfer and processing as a unit. In an embodiment, a pre-processor examines each class file in a set of class files to locate duplicate information in the form of redundant constants contained in a constant pool. The duplicate constant is placed in a separate shared table, and all occurrences of the constant are removed from the respective constant pools of the individual class files. During pre-processing, memory allocation requirements are determined for each class file, and used to determine a total allocation requirement for the set of class files. The shared table, the memory allocation requirements and the reduced class files are packaged as a unit in a multi-class file. *Fresko* does not teach or suggest "identifying a first data entity and a second data entity, wherein both the first data entity and the second data entity are identified using an Assembler instruction, and wherein the Assembler instruction identifies character strings which are required to appear only once in the executable file," as recited in amended independent claim 1.

In view of the above, Applicants respectfully submit that *Fresko* does not teach each and every feature of amended independent claim 1, as is required under 35 U.S.C § 102. In addition, *Pavlov* does not teach each and every feature of dependent claims 2, 5, 6 and 14 at least by virtue of their dependency on claim 1. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1, 2, 5, 6 and 14 under 35 U.S.C § 102.

II. 35 U.S.C. § 103, Obviousness

The Examiner has rejected claims 3 and 4 under 35 U.S.C. § 103 as being unpatentable over Fresko et al., U.S. Patent No. 5,966,702 (hereinafter "*Fresko*") in view of Ziedman, U.S. Patent Application Publication No. 2005/0114840 (hereinafter "*Ziedman*"). This rejection is respectfully traversed.

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). The prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In determining obviousness, the scope and content of the prior art are... determined; differences between the prior art and the claims at issue are... ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or non-obviousness of the subject matter is determined. *Graham v. John*

Deere Co., 383 U.S. 1 (1966). "Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue." *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). *"Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.* Id. (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006))."

Since claims 3 and 4 depend from independent claim 1, the same distinctions between *Fresko* and the invention recited in claim 1 apply to dependent claims 3 and 4. In addition, *Ziedman* does not provide for the deficiencies of *Fresko* with regard to amended independent claim 1. *Ziedman* is directed to a software tool for detecting plagiarism in computer source code. *Ziedman* is cited for teaching a "partial word matching" algorithm. *Ziedman* does not teach or suggest "identifying a first data entity and a second data entity, wherein both the first data entity and the second data entity are identified using an Assembler instruction, and wherein the Assembler instruction identifies character strings which are required to appear only once in the executable file," as recited in amended independent claim 1. Thus, any alleged combination of *Fresko* and with *Ziedman* still would not result in the invention recited in amended claim 1 from which claims 3 and 4 depend. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 3 and 4 under 35 U.S.C. § 103.

III. New Claims 21-22

In addition to being dependent on independent claim 1, claims 21 and 22 also distinguish over the *Fresko* and *Ziedman* references based on the specific features recited therein. With respect to claim 21, *Fresko* and *Ziedman*, taken alone or in combination, do not teach or suggest that "the Assembler instruction is a DL Assembler instruction." In addition, *Fresko* and *Ziedman*, taken alone or in combination, do not teach or suggest that "the DL Assembler instruction is a type of Assembler instruction that denotes a non-executable data entity which needs only be included once in the executable file," as recited in claim 22.

IV. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: November 24, 2008

Respectfully submitted,

/Gerald H. Glanzman/

Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

GHG/VJA